

AULA 4

BANCO DE DADOS

Banco de Dados com MySQL

Qual a diferença entre um dado e uma informação?

Enquanto os **dados** podem ser meros números, fatos não processados, a **informação** é o que dá sentido ao que foi analisado. Os **dados** não são específicos, ao contrário da **informação**, que é um significado detalhado a partir do trabalho com elementos mais brutos

O que é banco de dados?

O banco de dados é a **organização e armazenagem de informações sobre um domínio específico**. De forma mais simples, é o agrupamento de dados que tratam do mesmo assunto, e que precisam ser armazenados para segurança ou conferência futura.

Voltando um pouco no tempo...

Antigamente as empresas armazenavam dados em **fichas de papel** que eram organizadas em **arquivos físicos** através de **pastas**. Extrair informações e manter esses arquivos organizado era uma tarefa muito custosa. Além disso o acesso à informação dependia da localização geográfica dos arquivos.

Por isso, é interessante ter um sistema de gerenciamento de banco de dados, SGBD, para conseguir manipular as informações e tornar a rotina da empresa muito mais simples.

O que é um SGBD?

Sistema de Gerenciamento de Banco de Dados (SGBD) – Tudo que fazemos em um banco de dados passa pelo SGBD! O SGBD é responsável por tudo, salvar os dados, manter em memória os dados mais acessados, ligar dados e metadados, disponibilizar uma interface para programas e usuários externos acessem o banco de dados (para banco de dados relacionais, é utilizada a linguagem SQL), encriptar dados, controlar o acesso a informações, manter cópias dos dados para recuperação de uma possível falha, garantir transações no banco de dados, enfim, sem o SGBD o banco de dados não funciona!

Tipo de SGBD:

Hoje, existem diversos tipos de SGBDs, e cada um é adequado para uma necessidade dos clientes. São os mais comuns: Oracle, DB2, MySQL, SQL Server, PostgreSQL entre outros.

Todos os citados acima, utilizam a linguagem SQL como forma de consulta e manipulação do banco.

Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para banco de dados relacional.

Benefícios usar Banco de Dados:

- Melhora do relacionamento e produtividade na empresa
- Redução dos riscos de operação
- Aumento da segurança
- Melhora na tomada de decisão

Princípios básicos:

- Integridade: apenas dados reais e íntegros
- Segurança: Capacidade de assegurar os dados armazenados
- Tolerância a falhas
- Recuperação: Backup
- Controle de concorrência: Garante a atualização correta e simultânea de acesso.

Viabilizam a alteração, inclusão ou exclusão de dados de forma simplificada pelos clientes, que passam a ter altos níveis de satisfação e autonomia.

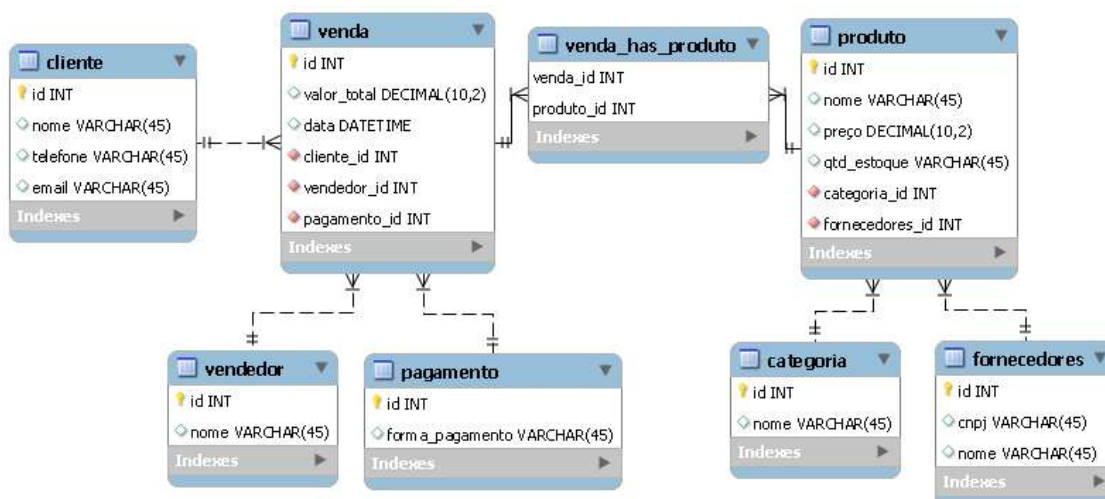
Como começar criar um banco de dados?

É recomendado que se inicie pela modelagem do banco de dados antes de sair escrevendo vários códigos, mas o que é a modelagem do banco de dados?

Modelagem de banco de dados é o processo de **levantamento, análise, categorização e exploração de todos os dados e tipos de informações** que irão sustentar uma aplicação. Esta é uma etapa primordial no trabalho do desenvolvimento de sistemas, porque todo software é criado com determinados objetivos, para atender às necessidades dos usuários dentro deste cenário.

Assim, se um sistema for desenvolvido sem que haja uma modelagem de banco de dados bem executada no início do projeto, **as chances de ele apresentar falhas ou até mesmo de não suprir os objetivos para os quais foi criado são grandes**. Por isso, atenção a esta etapa dentro do seu negócio antes de começar a desenvolver qualquer software.

Exemplo de uma modelagem de banco de dados:



Conhecendo os primeiros (e mais importantes) códigos SQL:

- INSERT: adicionando registros a uma tabela;
- UPDATE: atualizando os registros já inseridos;
- DELETE: excluindo registros de uma tabela;
- SELECT: retomando registros na tabela;
- CREATE: criando novas tabelas em um banco de dados;
- ALTER: alterando uma tabela já criada;
- DROP: excluindo uma tabela do banco de dados;

Os comandos acima, estão apenas com a primeira instrução para consulta de manipulação do banco de dados, antes de aprendermos a estrutura do comando por completo, devemos saber que cada comando desse possui um padrão lógico que chamamos de “sintaxe”.

A sintaxe nada mais é que seguir um padrão que o banco de dados entenda, e consiga processar a nossa instrução.

Por exemplo o seguinte comando para exibir dados do banco de dados:

```
INSERT INTO usuario (column1, column2, column3, ...)  
VALUES (value1, value2, value3, ...);
```

O comando mostrado acima está na sintaxe correta, mas caso nós alterarmos a sequência e primeiro informar os valores a serem inseridos e logo após as colunas do banco de dados, ao tentar executar o código, iremos obter um erro de “sintaxe”, pois não estamos seguindo os padrões e estruturas do comando corretamente, não obedecendo a sua sintaxe correta.

A seguir teremos os principais comandos SQL completos, com sua sintaxe correta da forma que devem ser utilizados.

Nos exemplos nós teremos algumas exemplificações do tipo: <lista_de_campos>, neste caso você deverá substituir esse exemplo, pelo nome dos campos da tabela criada previamente por você;

***NOTA: Os comandos mostrados abaixo são para Sistema de Gerenciamento de Banco de Dados (SGBD) MySQL.**

Vamos aos códigos:

CREATE DATABASE (criando banco de dados) A criação do banco de dados e suas tabelas é, muitas vezes, um dos primeiros passos da parte prática do desenvolvimento de um novo projeto.

```
CREATE DATABASE <nome_do_seu_projeto>
```

CREATE TABLE (criando tabelas)

```
CREATE TABLE fornecedores(  
codigo int(4) AUTO_INCREMENT,  
nome varchar(30) NOT NULL,  
email varchar(50),  
PRIMARY KEY (codigo)  
);
```

Agora vamos às explicações do que se tratam os comandos:

- **AUTO_INCREMENT** pode ser utilizado para automatizar um código que sirva de chave primária de uma tabela.
- **PRIMARY KEY** define a chave primária da tabela, isto é, o campo que serve como chave da tabela e que não pode ser repetido.
- **NOT NULL** define que um determinado campo seja de preenchimento obrigatório.

Agora já temos um banco de dados e uma tabela criada, com isso é possível manipular os dados do banco de dados.

SELECT simples

O comando **SELECT** permite recuperar os dados de uma tabela do banco de dados. A sintaxe mais básica do comando é:

```
SELECT <lista_de_campos>  
FROM <nome_da_tabela>
```

COMANDO WHERE

A cláusula Where permite ao comando SQL passar condições de filtragem. Veja o exemplo da **Listagem 1**.

```
SELECT CODIGO, NOME FROM CLIENTES  
WHERE CODIGO = 10
```

COMANDO INNER JOIN

A junção de tabelas no comando SELECT também pode ser feita com o comando JOIN. Este comando deve ser utilizado com a palavra reservada INNER:

- INNER: Semelhante ao uso do operador “=” na junção de tabelas. Aqui os registros sem correspondências não são incluídos. Esta cláusula é opcional e pode ser omitida no comando JOIN.

Para exemplificar, temos as tabelas abaixo:

```
SELECT A.CODIGO, A.DESCRICAO, B.DESCRICAO, B.QTD  
FROM PRODUTOS A  
INNER JOIN COMPONENTES B  
ON (A.CODIGO = B.CODPRODUTO)
```

INSERT INTO (inserir registros)

Esse exemplo cria um novo registro na tabela "Funcionários"

```
INSERT INTO Funcionários (Nome,Sobrenome, Título) VALUES  
("André", "Pereira", "Estagiário");
```

UPDATE (atualização de registros)

Para alterar um registro, usamos o comando `UPDATE`, com ele é possível editar os campos de sua tabela e colocar outro valor neles.

```
UPDATE fornecedores SET nome="Ricardo Arrigoni" WHERE codigo=1;
```

DELETE (deletar registros)

Ele é o responsável por remover todo e qualquer registro do banco de dados.

Nota: Uma vez executado, esse comando não é reversível, portanto, tome bastante cuidado ao deletar algum registro de seu banco de dados.

```
DELETE FROM fornecedores WHERE codigo=3;
```

Principais aplicações para gerenciamento de banco de dados com interface gráfica:

Normalmente, para o gerenciamento de [banco de dados](#), utilizamos o terminal para realizar tarefas rotineiras. Mas há quem prefira utilizar ferramentas com interface gráfica para gerenciar estas aplicações.

MySQL Workbench



SQL Server



phpMyAdmin



HeidiSQL



DBeaver

